PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-140236
(43)Date of publication of application : 17.05.2002

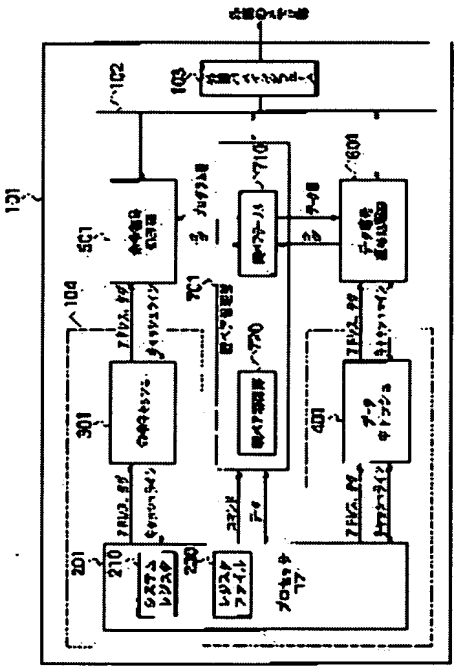
(51)Int.Cl. G06F 12/14
G06F 1/00
G06F 9/46
G09C 1/00

(21)Application number : 2000-333635
(22)Date of filing : 31.10.2000

(71)Applicant : TOSHIBA CORP
(72)Inventor : SHIRAKAWA KENJI
HASHIMOTO MIKIO
TERAMOTO KEIICHI
OZAKI SATORU
FUJIMOTO KENSAKU

(54) MICROPROCESSOR AND DATA PROTECTING METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To make contexts switchable by transfer of cipher data minimally necessary for maintaining secrecy pretection under a multitasking environment.
SOLUTION: This microprocessor has a command deciphering part deciphering an enciphered program using a first cipher key (a program key), and a data enciphering/deciphering part enciphering/deciphering data executed by the enciphered program using a second cipher key (a data key). A key pair managing part of the microprocessor has a key pair table storing the first and second keys in pairs. Tags specifying the key pairs are stored in a register file together with data related to the executed program. When a new program key is provided, a data key generating part of the key pair managing part generates a data key for enciphering/deciphering data executed by a program deciphered by the key. The pair of the keys is stored in the key pair table as a new key pair.



LEGAL STATUS

[Date of request for examination] 07.03.2005

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

(19)日本国特許庁（ＪＰ）　(12) 公開特許公報（A）　(11)特許出願公開番号
特開2002－140236
（P2002－140236A）
(43)公開日　平成14年５月17日(2002.5.17)

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
G06F	12/14	320	G06F	12/14	320B　5B017
	1/00			9/46	313A　5B076
	9/46	313	G09C	1/00	660D　5B098
G09C	1/00	660	G06F	9/06	660L　5J104

審査請求　未請求　請求項の数14　ＯＬ　（全 18 頁）

(21)出願番号　特願2000－333635(P2000－333635)

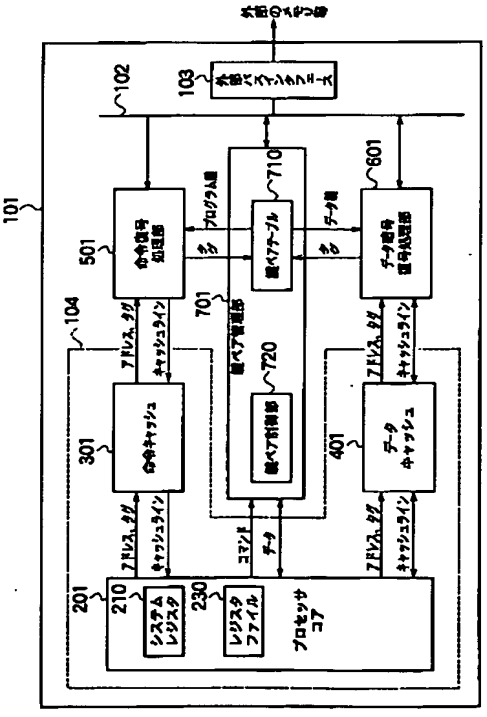
(22)出願日　平成12年10月31日(2000.10.31)

(71)出願人　000003078
株式会社東芝
東京都港区芝浦一丁目１番１号
(72)発明者　白川　健治
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(72)発明者　橋本　幹生
神奈川県川崎市幸区小向東芝町１番地　株式会社東芝研究開発センター内
(74)代理人　100083806
弁理士　三好　秀和　（外７名）

最終頁に続く

(54)【発明の名称】　マイクロプロセッサおよびデータ保護方法

(57)【要約】
【課題】　マルチタスク環境下で、秘密保護の保持に必要最小限な暗号データの転送でコンテキストの切り替えを可能にする。
【解決手段】　マイクロプロセッサは、暗号化されたプログラムを第1の暗号鍵(プログラム鍵)を用いて復号化する命令復号処理部と、復号化されたプログラムによって実行されていたデータを第2の暗号鍵(データ鍵)を用いて暗号化／復号化するデータ暗号／複号処理部を有する。マイクロプロセッサの鍵ペア管理部は、第1および第2の鍵をペアにして格納する鍵ペアテーブルを有する。この鍵ペアを特定するタグを、実行プログラムに関連するデータとともにレジスタファイルに格納する。鍵ペア管理部のデータ鍵生成部は、新たなプログラム鍵が与えられた場合に、この鍵で復号されたプログラムが実行するデータを暗号化／複号化するためのデータ鍵を生成する。この鍵のペアは、新規の鍵ペアとして鍵ペアテーブルに格納される。



1

【特許請求の範囲】

【請求項1】　暗号化されたプログラムを第1の暗号鍵を用いて復号化する命令復号処理部と、

前記復号化されたプログラムの実行対象であるデータを、第2の暗号鍵を用いて暗号化／復号化するデータ暗号／複号処理部と、

前記命令復号処理部およびデータ暗号／復号処理部に接続され、前記第1および第2の鍵を鍵ペアとして関連付けて格納する第1の記憶領域を有する鍵ペア管理部と、

前記鍵ペアを特定する識別子を、前記プログラムに関連するデータとともに格納する第2の記憶領域とを備えるマイクロプロセッサ。

【請求項2】　前記プログラムを復号するための第1の暗号鍵は、公開鍵暗号系で与えられ、

前記鍵ペア管理部は、前記第1の暗号鍵が与えられた場合に、当該第1の暗号鍵で復号されたプログラムによって実行されるデータを暗号化／複号化するための第2の暗号鍵を生成する鍵生成部をさらに備えることを特徴とする請求項1に記載のマイクロプロセッサ。

【請求項3】　前記マイクロプロセッサは、現在実行中のプログラムに用いられている有効な鍵ペアの識別子を格納する第3の記憶領域をさらに有し、

前記第3記憶領域の有効な鍵ペアの識別子の値が、特定の値を取る場合に、前記データ暗号／復号処理部は、前記第2の記憶領域に格納されていたデータを、そのデータに付随する識別子が指定する暗号鍵を用いて外部メモリに転送することを特徴とする請求項1に記載のマイクロプロセッサ。

【請求項4】　前記第1の記憶領域は、前記第1の鍵と第2の鍵を1対1対応で関連付けて格納することを特徴とする請求項1に記載のマイクロプロセッサ。

【請求項5】　前記第1の記憶領域は、前記第1の鍵のインデックスと、前記第2の鍵のインデックスとをペアにして格納する参照格納領域と、前記第1および第2の鍵を個別に格納する鍵格納領域とを含むことを特徴とする請求項1に記載のマイクロプロセッサ。

【請求項6】　前記マイクロプロセッサは、前記第2の記憶領域および第3の記憶領域に接続されるメモリアクセス部をさらに有し、

前記メモリアクセス部は、転送すべきデータに添付された鍵ペアの識別子と、前記第3記憶領域に格納されている現在有効な鍵ペアの識別子とに基づいてデータ転送の可否を判断するデータ転送判定部を有することを特徴とする請求項3に記載のマイクロプロセッサ。

【請求項7】　前記マイクロプロセッサは、前記第2の記憶領域および第3の記憶領域に接続される論理演算部をさらに有し、

前記論理演算部は、演算のオペランドに添付された識別子と、前記第3の記憶領域に格納されている有効な鍵ペアの識別子とに基づいて演算実行の可否を判断する演算

2

実行判定部を有することを特徴とする請求項3に記載のマイクロプロセッサ。

【請求項8】　前記第2の記憶領域は、複数のエントリから成り、各エントリが、プログラムに関連するデータと、そのデータに用いられる鍵ペアを特定する識別子とを有し、

前記第3記憶領域の有効な鍵ペアの識別子の値が特定の値をとる場合に、前記データ暗号／復号化処理部は、所望のエントリのデータおよび対応する識別子を暗号化して外部メモリに格納することを特徴とする請求項3に記載のマイクロプロセッサ。

【請求項9】　前記第3記憶領域の有効な鍵ペアの識別子は、例外発生時に前記特定の値をとることを特徴とする請求項3に記載のマイクロプロセッサ。

【請求項10】　前記鍵ペア管理部は、前記第1および第2の鍵のペアをプロセッサの秘密鍵で暗号化して格納する暗号化鍵ペア格納領域をさらに有することを特徴とする請求項2に記載のマイクロプロセッサ。

【請求項11】　暗号化されたプログラムを第1の鍵で復号するステップと、

前記第1の鍵に対応して、前記復号したプログラムによって実行されるデータを暗号化／復号化するための第2の鍵を生成するステップと、

前記第1の鍵と第2の鍵とを関連付けた鍵ペアとして格納するステップと、

前記鍵ペアに、該鍵ペアを特定する識別子を与えるステップと、

前記プログラムの実行中に例外が発生した場合に、前記識別子に基づいて前記第2の鍵を読み出し、前記データを第2の鍵で暗号化して外部のメモリに退避させるステップとを含むデータ保護方法。

【請求項12】　前記例外の終了後に、前記暗号化して退避させたデータ外部メモリから読み出して、前記識別子に基づき、前記第2の鍵で復号するステップをさらに含む請求項11に記載のデータ保護方法。

【請求項13】　前記鍵ペアとして格納するステップは、前記第1の鍵と第2の鍵とを1対1対応で直接組み合わせて格納することを特徴とする請求項11に記載のデータ保護方法。

【請求項14】　前記鍵ペアとして格納するステップは、前記第1の鍵のインデックスおよび第2の鍵のインデックスを生成し、第1の鍵のインデックスと第2の鍵のインデックスとをペアにして格納するステップを含むことを特徴とする請求項11に記載のデータ保護方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、マルチタスクのプログラム実行環境を支援するマイクロプロセッサと、このマイクロプロセッサによるデータ保護方法に関する。

【0002】

3

【従来の技術】計算機システムでプログラムを実行するにあたり、保護機能を持つプロセッサが求められている。現在実用化されているプロセッサにおいては、仮想メモリ管理おける保護機能や、周辺入出力デバイスへのアクセスの制約などの機構は、マルチタスクの実行環境を提供する計算機システムの安全性を確保するうえで、必須の条件とされている。

【０００３】さらに近年では、保護機構を考える上で、それ自体が保護されるべき一群のプログラムが特に重要な保護対象となってきている。たとえば、著作権により保護されているプログラムは、著作権を侵害する形で実行することが認められない。また、第三者から秘匿すべきデータを扱うプログラムもある。秘匿すべきデータを扱うプログラムが、その実行状態とともに解析される可能性があるとしたら、データの秘匿性が保証できなくなり、厳重な保護が要求される。

【０００４】これらプログラムを安全に実行するために、暗号学的に安全性を保証するシステムが提案され、実用化されている。その一つが、耐タンパ・ソフトウェアである(David Aucsmith et.al; ``Tamper Resistant Software: An Implementation'', Proceeding of the 1996, Intel Software Developer's Conference)。これは、プログラムの一部あるいは全部を暗号化して配布・保存し、これを利用する直前に自らプログラムとデータを復号化して実行し、終了時に必要であれば再び暗号化する技術である。

【０００５】しかし、耐タンパソフトウェア技術は、基本的に逆アセンブラ、デバッガなどの解析ツールによる解析を繁雑にするだけのものである。プログラムがプロセッサによって実行可能である以上、プログラムの開始時から順を追って解析していけば必ずプログラムの実行過程を解読分析することが可能である。つまり、プログラムの配布の過程においては、安全ではあっても、一旦実行するに至ると、実行する計算機システムにアクセスする手段をもつ第三者からは、プログラムとそのデータを秘匿することはできないものであった。

【０００６】ソフトウェアによって暗号・復号化を行なうのではなく、暗号化／複号化の処理機能自体をマイクロプロセッサに内蔵し、復号したプログラムをマイクロプロセッサの外部から隠匿するという一連の技術がある。これらは、Hampsonによる米国特許第4,47,902号、Hartmanの米国特許第5,224,166号、Davisの米国特許第5,806,706号、Takahasiの米国特許第5,825,878号、Leonard et.alの米国特許第6,003,117号、特開平11-282756号などに開示されている。

【０００７】マイクロプロセッサに暗号／複号処理機能を内蔵する方式に特徴的なことは、マイクロプロセッサが、秘密データを外部から物理的に隠蔽された形で保持できる点にある。すなわち、マイクロプロセッサの破壊的な検査によっても秘密が開示されない形で、秘密データが保持される。これを次のように利用する。公開鍵暗号系でいう秘密鍵を、秘密データとしてあらかじめマイクロプロセッサに内蔵させる。プログラムは、たとえばプログラムベンダーにより、ある暗号鍵(共通鍵暗号系を用いることが一般的である)で暗号化されて、マイクロプロセッサに配布される。このとき、暗号鍵を、プロセッサの秘密鍵に対応する公開鍵でさらに暗号化してプログラムに添付する。

4

【０００８】この機構により、プログラム自体を安全な形で提供することが可能であり、その実行過程を逆アセンブルなどの解析手法を用いて復元することはできない。また、プログラムの暗号鍵を知ることなくプログラムを意図する形に改変することは、暗号学的に困難である。

【０００９】しかし、暗号プログラムを実行するこれらのマイクロプロセッサには、大きな問題点があった。それは、これらマイクロプロセッサは、通常、マルチタスク環境を実現するオペレーティング・システム(以下、「ＯＳ」という)のもとで使用されることである。マルチタスク環境においては、コンテキスト・スイッチという操作でマイクロプロセッサを時分割することで、複数のプログラムを見かけ上同時に実行する。このコンテキスト・スイッチの過程において、ＯＳは、プロセッサの実行状態の全てを読み書きすることが可能である。したがって、ＯＳの挙動を解析すること、あるいはＯＳ自体を改竄することにより、暗号で保護されているはずのプログラムも、その実行状態が全て解析されるおそれがある。

【００１０】この問題に対して、従来技術は部分的な解決策を提案している。たとえば、特開平第11-282756号公報は、アプリケーションの持つ秘密情報を保存するためにプロセッサ内に設けられた秘密メモリの技術を開示している。この例では秘密メモリの中のデータへアクセスするために、予め定められた参照値を必要とする。しかしながら、同一のプロセッサで複数のプログラムが走っている場合に、特にオペレーティング・システムから、秘密データのアクセス権を得るための参照値をどのように保護するかは、なんら開示されていない。

【００１１】また、特許出願番号第２０００－１３５０１０号公報に開示される技術では、コンテキスト・スイッチをハードウェアで行ない、そのときにレジスタファイルの内容を暗号化した上でメモリに一時的に退避することにより、オペレーティング・システムからプログラムの実行状態を隠匿している。これにより、メモリに退避したプロセッサの実行状態を解析して秘密データを復元することは暗号学的に困難となり、暗号プログラムの安全性を保証することができる。しかしながら、この技術にも、次のような問題がある。

【００１２】

【発明が解決しようとする課題】第1の問題は、暗号に

5

よる保護の対象がレジスタファイルに限定されている点である。マイクロプロセッサ内部に、レジスタファイル以外に多数あるいは大容量のメモリを持つ場合も想定され、その場合、コンテキスト・スイッチングの負荷は容量に比例して大きくなると思われる。しかし上記公報には、これら内部メモリをどのように保護するか、あるいはコンテキスト・スイッチの負荷をいかに軽減するかは、まったく開示されていない。

【００１３】たとえばプロセッサが大容量の内部メモリを持つ場合、従来の技術では、メモリのすべてを保護できないため、保護データの漏洩は避けられない。また、すべてを保護しようとすると、大容量のため暗号処理に伴う性能劣化を招き、実用化する上での大きな制約となっている。

【００１４】大容量のキャッシュ・メモリを持つプロセッサにおいて、コンテキスト・スイッチの負荷を軽減させるために用いられる手法に、キャッシュの連想メモリ部でのヒット・ミスヒットの判定に用いるタグに、キャッシュされているデータを所有するプロセスを特定する識別子を含ませるという方法が知られている(以下、プロセス・タグ方式と呼ぶ)。

【００１５】しかし、この方法を単純に適用することは難しい。プロセスとは、オペレーティング・システムがプロセッサを仮想化するために用いる概念であり、プロセッサそのものがプロセスを特定する識別子を扱うことは困難だからである。また、仮に、プロセッサがプロセスを特定する識別子によってデータへのアクセスを保護する仕組みを持つとしても、プロセス識別子がオペレーティング・システムの管理下にある以上、オペレーティング・システムの改竄による秘密の漏洩に対しては、とるべき措置がない。

【００１６】第2の問題点は、コンテキストのスイッチングがハードウェアで固定されており、すべてのレジスタが保存あるいは復帰されることになり、柔軟性に欠ける点である。たとえば、頻繁に起きる例外の処理のときには、レジスタファイルの一部だけを保存・復帰するといった最適化が望まれるが、特許出願番号第２０００－１３５０１０号公報に開示される技術では、レジスタファイルの内容がすべて一括して保存・復帰される。

【００１７】そこで、本発明の第1の目的は、プログラム自体の秘密と、プログラムが扱うデータの秘密の双方を暗号学的に保証するとともに、マルチタスク環境下でコンテキスト・スイッチの負荷を軽減させることのできるマイクロプロセッサを提供することにある。

【００１８】本発明の第2の目的は、マルチタスク環境下で、必要な部分だけのデータの保存・復帰を可能にするために、保護対象の処理の最適化を行なうことにある。

【００１９】

【課題を解決するための手段】上記第1および第2の目

6

的を同時に達成するために、本発明では、基本的に、公開鍵暗号系によるプログラムの暗号化鍵の配布方式を前提とし、公開鍵暗号系で与えられる、暗号プログラムを復号するためのプログラム鍵(第1の鍵)と、このプログラムによって処理されたデータを暗号化／複号化するためにマイクロプロセッサが生成するデータ鍵(第2の鍵)とを、鍵ペアとして関連付けて保持する鍵管理方式を提案する。この方式によれば、マイクロプロセッサは、暗号化されたプログラムを第1の鍵を用いて復号化し、第1の鍵に対応する第2の鍵を生成するとともに、これらの鍵の組み合わせに対して識別子を与える。第1の鍵と第2の鍵は、鍵ペアとして鍵ペアテーブルに書き込まれ、前記識別子はこの鍵ペアを特定するために以降の処理で使用される。復号化されたプログラムの実行中には、このプログラムの復号に用いた第1の鍵の鍵ペアに与えられた識別子を提示しておく。割り込みなどにより、復号されたプログラムの実行が中断される場合は、このプログラムによって処理されていたデータを、現在の識別子に対応する第2の鍵で暗号化し、外部のメモリ等に退避させる。処理されたデータに対するアクセス権の判定には、鍵ペアの一致を用いることで、プロセス間のデータの保護を暗号学的に保証する。

【００２０】このような方式を実現するために、本発明に係るマイクロプロセッサは、暗号化されたプログラムを第1の暗号鍵(たとえばプログラム鍵)を用いて復号化する命令復号処理部と、復号化されたプログラムによって実行されていたデータ(すなわちプロセスの実行状態)を第2の暗号鍵(たとえばデータ鍵)を用いて暗号化／復号化するデータ暗号／複号処理部と、第1および第2の鍵をペアにして格納する第1の記憶領域(たとえば鍵ペアテーブル)を有する鍵ペア管理部と、この鍵ペアを特定するタグ(識別子)を、前記プログラムに関連するデータとともに格納する第2の記憶領域(たとえばレジスタファイル)とを有する。

【００２１】プログラム鍵とデータ鍵のペアは不可分に格納され、これをプログラムの側から操作するときには、鍵ペアをプロセッサに固有の秘密鍵で暗号化した形でのみ操作される。これにより、暗号化プログラムの実行にあたり、プログラム自体の解析を暗号学的に困難とし、かつプログラムの実行状態の解析をオペレーティング・システムからも困難とする。また、鍵ペアを特定するタグを対応する各データに添付することによって、暗号処理を伴うデータ転送を、秘密保持に必要最小限の範囲で行なうことができる。

【００２２】暗号化プログラムを復号するためのプログラム鍵は、公開鍵暗号系で与えられる。マイクロプロセッサは、データ鍵生成部をさらに有し、新たなプログラム鍵が与えられた場合に、このプログラム鍵で復号されたプログラムによって実行されるデータを暗号化／複号化するためのデータ鍵を生成する。このようにして生成

された鍵のペアは、鍵ペアテーブルに格納される。

【００２３】マイクロプロセッサは、現在実行中のプログラムに用いられている有効な鍵ペアの識別子である、実効鍵ペア識別子を格納する第3 の記憶領域(システムレジスタ) を有する。プログラムが、ユーザモードで実行しているときに、割り込みなどによりカーネルモードに遷移すると、実効鍵ペア識別子は、カーネルモードを示す特定の値に切り替わる。これにより、ユーザモードのプログラムと、割り込み処理プログラムとを明確に区別することができる。割り込み処理プログラムが、ユーザモードのプログラムによって第2 の記憶領域に納められていたデータを外部に一時的に退避させるにあたって、データ暗号／復号処理部は、そのデータに付随する識別子が指定する暗号鍵を用いて、外部メモリに転送する。これにより、割り込み等による例外発生時にも、データの安全は保護される。

【００２４】鍵ペアテーブルは、第1 の鍵(プログラム鍵) と第2 の鍵(データ鍵) を1 対1 対応でペアにして、複数の鍵ペアを格納する。

【００２５】あるいは、鍵ペアテーブルは、第1 の鍵のインデックスと、第2 の鍵のインデックスとを関連付けて格納する参照格納領域と、第1 および第2 の鍵を個別に格納する鍵格納領域とを含む構成としてもよい。この場合、鍵のインデックス自体はサイズが小さく、参照領域の記憶容量は少なくてすむ。また、個別に格納された第1 および第2 の鍵は、インデックスによって指定されることになるので、たとえば、マルチタスク環境下で、同じひとつのプログラムに対して異なる種類のデータが処理されている場合など、インデックスを用いて、プログラムの鍵と、処理データの鍵とを適正に組み合わせて複数の鍵ペアを特定することが可能になる。

【００２６】マイクロプロセッサは、第2 の記憶領域と第3 の記憶領域に接続されるメモリアクセス部をさらに有する。メモリアクセス部は、転送すべきデータに添付された鍵ペアの識別子と、実効鍵ペア識別子とに基づいてデータ転送の可否を判断するデータ転送判定部を有する。

【００２７】マイクロプロセッサはまた、第2 の記憶領域と第3 の記憶領域に接続される論理演算部をさらに有する。論理演算部は、演算のオペランドに添付された識別子と、実効鍵ペア識別子とに基づいて演算実効の可否を判断する演算実効判定部を有する。

【００２８】このように、鍵ペアを特定する識別子を、マイクロプロセッサ内で扱うデータに付加し、データ転送や演算操作の際に、データに付随する鍵ペアの識別子をアクセス権や演算可否の判断に用いることによって、データの安全性を一層向上することができる。

【００２９】第2 の記憶領域は、複数のエントリから成り、各エントリが、プログラムに関連するデータと、そのデータに用いられる鍵ペアを特定する識別子とを有する。この構成により、たとえば割り込みによりカーネルモードでの処理が希望され、第3 記憶領域の実効鍵ペア識別子の値がカーネルモードを示す値を取った場合に、所望のエントリのデータおよび対応する識別子だけを暗号化して、外部メモリに退避させることが可能になる。すなわち、割り込み発生時に、第2 記憶領域内にあるデータ全体の退避に加えて、その一部だけを暗号化して退避させることが可能になる。

【００３０】本発明のその他の特徴、効果は、以下の詳細な説明でいっそう明確になる。

【００３１】

【発明の実施の形態】＜第1 実施形態＞図1から図１５を参照して、本発明の第一実施形態にかかるマイクロプロセッサを説明する。本発明において、マイクロプロセッサは、暗号化プログラムをマルチタスク環境下で実行することを前提とする。

【００３２】図1は、本発明の第一実施形態に係るマイクロプロセッサ１０１の機能構成を示すブロック図である。マイクロプロセッサ１０１は、プログラムを実行するプロセッサ・コア２０１、プログラムの命令列を一時的に格納する命令キャッシュ３０１、このプログラムによって処理されるデータを一時的に格納するデータキャッシュ４０１、暗号化されたプログラムを実行時に復号してプロセッサコアに供給する命令復号処理部５０１、復号されたプログラムによって実行されていたデータを暗号化あるいは復号化するデータ暗号／複号処理部６０１と、鍵ペア管理部７０１を有する。

【００３３】命令復号処理部５０１において、暗号化プログラムの復号には、公開鍵系で与えられるプログラム鍵を用いる。また、プログラムが実行していたデータの暗号化／複号化には、プログラム鍵に対応して生成されたデータ鍵を用いる。第1 実施形態の特徴として、鍵ペア管理部７０１は、このプログラム鍵とデータ鍵とを1 対1 対応でペアにして格納する鍵ペアテーブルを有するが、これについての詳細は後述する。

【００３４】また、プロセッサ・コア２０１は、システムレジスタ２１０と、レジスタファイル２３０を含む。システムレジスタ２１０は、現在実行中のプログラムのための鍵ペアを特定するタグ(識別子) を提示する。レジスタファイルは、プログラムデータあるいは処理データを、それに対応する鍵ペアのタグとともに格納する。これらの機能の詳細については後述する。

【００３５】マイクロプロセッサ１０１はさらに、プロセッサ・バス１０２、外部バス・インタフェース１０３を有し、マイクロプロセッサの外部にあるメモリや周辺デバイスと外部バス・インタフェース１０３を介して接続される。

【００３６】プロセッサ・コア２０１、命令キャッシュ３０１、データ・キャッシュ４０１は、図1 において破線１０４で示される保護領域内にある。マイクロプロセ

ッサ101の保護領域104は、外部あるいはOSから保護される領域であり、この領域内部ではデータは平文状態で取り扱われる。一方、保護領域104の外部では、秘匿すべきデータは必ず暗号化されている。保護領域外部から、暗号化されたデータが保護領域104の内部に読み込まれるときに、それが命令として読み込まれるときには命令復号化部501で復号化され、データとして読み込まれるときには、データ暗号復号化部601で復号化される。復号化にあたって使用するプログラム鍵およびデータ鍵は、鍵ペア管理部701から供給される。

【0037】上述したように、第1実施形態の特徴として、復号化された平文状態のデータには、暗号操作を経たことを示す属性として、復号処理に用いた鍵ペアを特定するタグが付加され、プロセッサコア201内のレジスタファイル230に格納される。

【0038】図2および図3は、マイクロプロセッサ101の処理操作の大まかな流れを示す。

【0039】まず、図2に示すように、ステップS21で、マイクロプロセッサ101の外部のメモリに格納された暗号化プログラムの命令列を、現在実行中のプログラムの有効な鍵ペア（以下、「実効鍵ペア」と称する）のプログラム鍵を用いて復号化する。現在の実効鍵ペアは、システムレジスタに鍵ペアのタグが提示されているので、このタグに基づいて知ることができる。図2の例では、実効ペアタグの値は#1である。このタグ値に基づいて、鍵ペア管理部の鍵ペアテーブルからタグ#1に対応するプログラム鍵を読み出す。

【0040】次に、ステップS23に示すように、復号化されて平文状態になった命令列にしたがって、プログラムを実行する。このプログラムの実行、すなわち演算操作の結果得られたデータには、鍵ペアタグが添付されて、レジスタファイル230に格納される。

【0041】次にステップS25に示すように、レジスタファイルのデータをいったんデータキャッシュ401に転送する。

【0042】最後に、ステップS27に示すように、このデータに添付された鍵ペアタグに基づいて、鍵ペアテーブルからデータ鍵を取りだし、データ鍵でデータを暗号化する。暗号化されたデータを外部メモリに転送する（退避させる）。

【0043】図3は、退避させたデータの復帰処理フローである。まず、復帰するにあたって使用するべき鍵ペアタグを特定する。そして、外部メモリから暗号化されたデータをマイクロプロセッサ内部に取り込む。鍵ペアタグが指定するデータ鍵を鍵ペアテーブルから取り出し、これを用いてデータを復号し、保護領域の内にあるデータキャッシュ401にキャッシュする。

【0044】次に、ステップS33に示すように、データキャッシュ401上の平文データをレジスタファイル230に転送する。そして、ステップS35に示すように、レジスタファイル230上のデータに対する演算操作を再開する。

【0045】図4から図11は、このような操作を行なうマイクロプロセッサ101の各構成要素の詳細図である。これらの図に基づいて、各要素の構成と、鍵ペア・タグによる保護機能について説明する。

【0046】図4は、プロセッサ・コア201の詳細な構成例を示す。以下の説明においては、MIPS Technologies社のRISC型マイクロプロセッサのアーキテクチャに、本発明に基づく変更を加えたものを例にとって説明する。より具体的には、プロセッサのパイプライン構成については、MIPS Technologies社のR3000系のものを、命令セットについては、MIPS-IないしはMIPS-IV命令セットを例にとって説明するが、これは本発明の適用範囲をMIPS Technologies社のプロセッサに限定するものではない。

【0047】なお、図4のプロセッサ・コア201の5つのパイプラインステージIF（instruction fetch）、RF（register read）、EX（execution）、MEM（memory access）、WB（write back）を左端に示す。

【0048】プロセッサ・コア201は、システムレジスタ210と、命令フェッチ・デコーダ220と、レジスタファイル230と、演算部250と、メモリアクセス部260を含む。

【0049】システムレジスタ210は、MIPS R3000のCP0に相当するシステム・レジスタをもとにして、仮想アドレス管理や例外処理機能に追加して、現在実行しているプロセスを特定する有効な鍵ペアのタグを格納する実効鍵ペアタグ・レジスタ211を備える。すなわち、実効鍵ペアタグ・レジスタ211は、現在実行中の有効なプログラムと、このプログラムによって処理されているデータとを暗号処理する鍵のペアを示すタグを格納している。

【0050】命令フェッチ・デコーダ220は、プログラムカウンタ221と、命令バッファ222と、命令実行制御部223を含む。命令実行制御部223の制御のもとに、プログラムカウンタ221の指すアドレスから、命令を命令バッファ222にフェッチし、これをデコードして、各データ・パスを制御する信号(図中では省略)を生成する。

【0051】第1実施形態では、命令キャッシュ301への読み出し要求のパラメータとして、命令アドレスに加えて、現在実行中のプロセスを示す実効鍵ペアタグ・レジスタ211の値も送られる。

【0052】レジスタファイル230は、各レジスタ231において、レジスタデータ部231－1とともに、本発明に特徴的なレジスタタグ部231－2を持つ。レジスタタグ部231－2は、そのレジスタに格納された

データの暗号による保護属性を示す鍵ペア・タグが格納される。

【００５３】命令実行パイプラインのＲＦ（レジスタ・リード）フェーズでは、レジスタデータ部２３１－１の内容がオペランド・バス２４０に、レジスタタグ部２３１－２の内容がオペランドタグ・バス２４１に置かれる。

【００５４】算術論理演算部２５０は、命令実行パイプラインのＥＸフェーズに相当する。第1 実施形態では、算術論理演算部２５０は、通常の演算器（演算データパス）２５１の他に、演算オペランドタグ判定部２５２を備える。図5 に示すように、演算オペランドのタグ判定部２５２は、少くとも、演算の種類、演算のオペランドに付加されたタグの値、および実効鍵ペア・タグの三つの値に応じて、演算の実行の可否を決定する機能を持つ。演算の種類は、実効制御部２２３からタグ判定部２５２に入力され、タグ値は、レジスタファイルのタグ部２３１－２から入力される。現在の実効鍵ペア・タグは、システムレジスタ２１０の実効鍵ペア・タグ・レジスタ２１１から入力される。タグ判定部２５２によって実行が否決された場合には、プロセッサ・コア２０１は、再開のできない例外を発生し、その命令はアボートされる。実行が可決された場合には、命令に実行結果が存在すれば、その結果のデータがリザルト・バス２８０に、結果の鍵ペア・タグがリザルトタグ・バス２８１に置かれる。

【００５５】メモリアクセス部２６０は、命令実行パイプラインのＥＸ／ＭＥＭフェーズに相当する。第1 実施形態では、メモリアクセス部２６０は、通常のアドレス計算部２６１の他に、データ転送タグ判定部２６２を有する。メモリアクセス部２６０は、データ・キャッシュ４０１への読み書き要求のパラメータとして、アドレス計算部２６１で求めたデータ・アドレスに加えて、オペランドに付加されたタグの値あるいは実効鍵ペア・タグの値の、いずれか一方も送出する。データ転送タグ判定部２６２は、少くとも、データの転送元、データの転送先、転送するデータに付加されたタグの値、および実効鍵ペア・タグの四つの値に応じて、転送の実行の可否を決定する機能を持つ。実行が否決された場合には、プロセッサ・コアは、再開のできない例外を発生し、その命令はアボートされる。実行が可決された場合には、命令に実行結果が存在すれば、その結果のデータがリザルト・バス２８０に、結果の鍵ペア・タグがリザルトタグ・バス２８１に置かれる。

【００５６】鍵ペア管理部インタフェース２７０は、本発明に固有のものであり、後述する鍵ペア管理部７０１を制御するためのものである。

【００５７】命令実行パイプラインのWBステージでは、リザルト・バス２８０上のデータと、リザルトタグ・バス２８１上の鍵ペア・タグが、必要であれば、レジスタファイル２３０に書き戻される。

【００５８】図6 は、現在の実効鍵ペア・タグを提示するシステムレジスタ２１０の構成図である。実効鍵ペア・タグ・レジスタ２１１は、現在実行中のプログラムの有効な鍵ペアのタグを提示する。第1 実施形態では、鍵ペア・タグのうち、２つのタグ値を、特別な用途のために予約している。

【００５９】一つは、暗号処理を行わないことを示すためのタグ値である（これをゼロ・タグと呼ぶ）。命令復号処理部５０１と、データ暗号処理部６０１においては、鍵ペア・タグとしてゼロ・タグ（Tag-0）が提示されると、暗号操作を行わずに、データ（あるいは命令）をそのまま転送する。

【００６０】もう一つは、プロセッサの動作モードとして、カーネルモードが選択されている時に用いるために予約されているタグ値（tag-K）である。このタグ値に対応する鍵ペアには、カーネルモードで実行するプロセス（通常の計算機システムにおいては、オペレーティング・システム）のプログラム鍵およびデータ鍵が登録される。

【００６１】実効鍵ペア・タグ・レジスタ２１１は、カーネルモードでの鍵ペア・タグを格納するカーネル鍵ペアタグ・レジスタ２１１－1と、ユーザモードでの鍵ペア・タグを格納するユーザ鍵ペアタグ・レジスタ２１１－2を備え、そのときのプロセッサの実効モードによりいずれかが選択されて、以下の処理で有効な鍵ペアタグ（実効鍵ペアタグ）として出力される。

【００６２】図7 は、命令キャッシュ３０１の詳細な構成例を示す。命令キャッシュ３０１は、複数の命令キャッシュライン３０２の配列で構成される。本発明のキャッシュラインは、キャッシュ中のアドレスの検索方法は公知の手法で行うが、本発明に特徴的な要素として、各命令キャッシュラインは、そのラインにキャッシュされているプログラムデータの保護属性を示す、すなわちキャッシュされているプログラムデータに作用させるべきプログラム鍵を示す鍵ペア・タグを格納する領域３０２－1を有する。鍵ペア・タグ領域３０２－1の大きさは、後述する鍵ペアテーブルの全エントリをインデックスできるだけのビット数があればよい。たとえば６４エントリの鍵ペアテーブルを用いるとすれば、６ビットあれば十分である。なお、プログラムデータの外部メモリでの所在を示すアドレスや、その状態を示す領域も、各キャッシュラインに設けられている。

【００６３】図8 は、データ・キャッシュ４０１の詳細な構成例を示す。データキャッシュ４０１は、複数のデータキャッシュライン４０２の配列で構成される。データキャッシュラインもまた、そのラインにキャッシュされている処理データの保護属性を示す、すなわちキャッシュされている処理データに作用させるべきデータ鍵を示す鍵ペア・タグを格納する領域４０２－1を有する。

外部メモリのアドレスおよび状態を示す領域が設けられているのは、命令キャッシュ３０１と同様である。

【００６４】図9は、命令復号処理部５０１の詳細な構成例とその動作を示す。命令復号処理部５０１は、復号化処理の対象となっているプログラムデータやその暗号鍵を一時的に保持するコマンドデータ・レジスタ５０２と、共通鍵による復号化を行う命令復号演算器５０３と、これらを制御するための命令復号制御部５０４とから構成される。

【００６５】命令復号処理部５０１はまず、命令キャッシュ３０１から読み出し要求を受ける。このときのパラメータは、外部メモリのアドレスと、読み出したプログラムデータに対して作用させる暗号鍵(プログラム鍵)を特定する鍵ペア・タグである。まず、アドレスをパラメータとして外部メモリに読み出し要求を出す。また、鍵ペア・タグをパラメータとして、鍵ペア管理部７０１にプログラム鍵の読み出し要求を出す。

【００６６】これらの読み出し要求に応じて外部メモリから送られてきた暗号化プログラムデータとプログラム鍵は、コマンドデータ・レジスタ５０２に格納される。命令復号演算器５０３は、コマンドデータ・レジスタ５０２上の暗号化されているプログラムデータに対してプログラム鍵を作用させ、暗号プログラムデータを復号化する。復号化が完了すると、平文のデータが命令キャッシュへと出力される。

【００６７】図１０は、データ暗号処理部６０１の詳細な構成例とその動作を示す。データ暗号処理部６０１は、プログラムによって処理された平文状態あるいは暗号状態のデータを一時的に格納するコマンドデータ・レジスタ６０２と、共通鍵による暗号化／復号化を行なうデータ暗号／復号演算器６０３と、これらを制御するためのデータ暗号制御部６０４とから構成される。

【００６８】データ暗号処理部６０１は、データキャッシュ４０１から読み出しおよび書き込み要求を受ける。読み出し要求は、たとえば割り込みによる例外発生時に、一時的に外部のメモリに退避させておいたデータを、割り込み処理後に呼び戻す場合に、発せられる。書き込み命令は、割り込み発生時に、それまで処理していたデータを護るためにいったん暗号化して外部のメモリに書き込む場合に発せられる。

【００６９】読み出し要求のパラメータは、外部メモリのアドレスと、読み出したデータに対して作用させる暗号鍵を特定する鍵ペア・タグである。まず、アドレスをパラメータとして外部メモリに読み出し要求を出す。また、鍵ペア・タグをパラメータとして鍵ペア管理部７０１にデータ鍵の読み出し要求を出す。外部メモリに暗号化された状態で格納されていた処理データと、鍵ペア管理部７０１からのデータ鍵は、コマンドデータ・レジスタ６０１に格納される。データ暗号演算器６０２は、コマンドデータ・レジスタ６０２上のデータに対してデータ鍵を作用させ、暗号化されていたデータを復号化する。復号化が完了すると、平文のデータがデータ・キャッシュへと出力される。

【００７０】一方、書き込み要求のパラメータは、処理データを書き込むべき(すなわち一時的に退避させるべき)外部メモリのアドレスと、転送するデータと、データに対して作用させる暗号鍵を特定する鍵ペア・タグである。まず、鍵ペア・タグをパラメータとして鍵ペア管理部７０１にデータ鍵の読み出し要求を出す。データ暗号／復号演算器６０３は、コマンドデータ・レジスタ６０２上の平文データに対してデータ鍵を作用させ、平文データを暗号化する。暗号化が完了すると、暗号化されたデータが外部メモリへと出力される。

【００７１】図１１は、鍵ペア管理部７０１の詳細な構成例を示す。鍵ペア管理部７０１は、プロセッサ・コア２０１とのインタフェース７０２と、命令復号処理部５０１とのインタフェース７０３と、データ暗号処理部６０１とのインタフェース７０４と、鍵ペア・テーブル７１０と、鍵ペア制御部７２０とから構成される。

【００７２】鍵ペア・テーブル７１０は、複数の鍵ペアのエントリー７１１を有する。それぞれの鍵ペアは、プログラム鍵７１１－１とデータ鍵７１１－２とから構成される。本発明でいう鍵ペア・タグとは、この鍵ペアの配列からなる鍵ペア・テーブル７１０のインデックスである。鍵ペア・テーブル７１０の動作は、以下で説明する三つの読み出し操作と、一つの書き込み操作である。

【００７３】(１)プログラム鍵の読み出し
命令復号処理部５０１と鍵ペア管理部７０１との間で行われる動作であり、命令復号処理部インタフェース７０３と接続されるポートを介する読み出し操作である。命令復号処理部５０１のレジスタにある鍵ペア・タグ(すなわちインデックス)によって特定される鍵ペアのプログラム鍵を取り出して出力する。

【００７４】(２)データ鍵の読み出し
データ暗号／復号処理部６０１と鍵ペア管理部７０１との間で行われる動作であり、データ暗号処理部インタフェース７０４と接続されるポートを介する読み出し操作である。データ暗号／復号処理部６０１のレジスタにある鍵ペア・タグ(すなわちインデックス)により特定される鍵ペアのデータ鍵を取り出して出力する。

【００７５】(３)鍵ペアの読み出し
プロセッサ・コア２０１と鍵ペア管理部７０１との間で行われる動作であり、鍵ペア制御部７１０と接続されるポートを介する読み出し操作である。鍵ペア・タグにより特定される鍵ペアのプログラム鍵とデータ鍵の両方を出力する。

【００７６】(４)鍵ペアの書き込み
鍵ペア制御部７２０と接続されるポートを介する書き込み操作であって、パラメータとして与えられるプログラム鍵とデータ鍵とを、インデックスにより指定される鍵

ペアテーブル上の鍵ペアとして格納する。

【００７７】鍵ペア管理部７０１の鍵ペア制御部７２０は、プロセッサ・コア２０１からの要求に従い、次の三つの操作を行なう。

【００７８】（１）新しい鍵ペアの登録

新規の暗号プログラムを実行する場合、その暗号プログラムを復号するためのプログラム鍵と、このプログラムによって実行処理されたデータを暗号化／復号化するためのデータ鍵をペアにして新規に登録する必要がある。鍵ペア制御部７２０は、プロセッサ・コア２０１から、プログラム鍵をプロセッサの公開鍵で暗号化したプログラム鍵データと、これに用いられることになる鍵ペア・タグを提示される。プログラム鍵データは、鍵登録用レジスタ７２１に格納され、一方、鍵ペア・タグは鍵ペア・テーブル７１０へのインデックスとして用いられる。

【００７９】公開鍵暗号処理部７２２は、プロセッサの秘密鍵を用いて、鍵登録用レジスタ７２１上の鍵データを復号化し、鍵ペア・レジスタ７２４のプログラム鍵格納領域７２４－１に格納する。また、データ鍵生成部７２３は、任意の手段、たとえば乱数発生機能でデータ鍵を生成し、これを鍵ペア・レジスタ７２４のデータ鍵格納領域７２４－２に格納する。鍵ペア制御部７２０は、これら二つの鍵がレジスタ７２４に格納されると、鍵ペア・テーブル７１０の鍵ペア書き込み操作に基づいて、鍵ペア・テーブル７１０に新規の鍵ペアを登録する。

【００８０】（２）既存の鍵ペアの読み出し

鍵ペア制御部７２０は、プロセッサ・コア２０１から、必要な鍵ペア・タグを提示する。この鍵ペア・タグは鍵ペア・テーブル７１０へのインデックスとして用いられる。鍵ペア・テーブル７１０は、その読み出し操作によって、インデックスが指定する鍵ペアを取り出し、その結果を鍵ペア制御部７２０の鍵ペア・レジスタ７２４に格納する。鍵ペア暗号処理部７２５は、鍵ペア・レジスタ７２４上に格納されている平文状態のプログラム鍵７２４－１とデータ鍵７２４－２とを、一つのデータとしてプロセッサの秘密鍵を用いて暗号化し、その結果を暗号化鍵ペア・レジスタ７２６に格納する。鍵ペア制御部７２０は、暗号化鍵ペア・レジスタ７２６上のデータをプロセッサ・コアへの出力とする。

【００８１】（３）既存の鍵ペアの書き込み

鍵ペア制御部７２０は、プロセッサ・コア201から、鍵ペア・タグ、およびプロセッサの秘密鍵で暗号化された鍵ペアである鍵ペア・データが提示される。鍵ペア・データは暗号化鍵ペア・レジスタ７２６に格納され、鍵ペア・タグは鍵ペア・テーブル７１０へのインデックスとして用いられる。 鍵ペア暗号処理部７２５は、暗号化鍵ペア・レジスタ７２６上のデータを、プロセッサの秘密鍵を用いて復号化する。結果として得られた平文データは、プログラム鍵およびデータ鍵の2つの鍵として、鍵ペア・レジスタ７２４に格納される。鍵ペアテーブル７１０は、その書き込み操作に基づいて、レジスタ７２４上の鍵ペアを書き戻す。

【００８２】本発明のマイクロプロセッサには、内部にデータを格納するためのメモリが複数存在する。さらに、外部メモリ・インタフェース１０３を介してアクセスされる外部メモリが存在する。これらのうち、マイクロプロセッサの内部にあって、鍵ペア・タグを格納する領域が付加されているメモリ（たとえばシステムレジスタ２１０、レジスタファイル２３０など）を、「内部メモリ」と呼ぶ。内部メモリのうち、さらにこれがキャッシュ・メモリ（たとえば命令キャッシュ３０１、データキャッシュ４０１など）である場合には、「内部キャッシュ・メモリ」と呼ぶ。プロセッサの外部にあるメモリ、あるいは、プロセッサの内部にあるが鍵ペア・タグを格納する領域を持たないメモリは、「外部メモリ」と呼ぶ。

【００８３】以下で、これらのメモリの間でのデータ転送の詳細について述べる。メモリ間の転送には、転送元と転送先が、内部であるか外部であるかによって、(i)内部メモリから内部メモリへ、(ii)内部メモリから外部メモリへ、(iii)外部メモリから内部メモリへ、(iv)外部メモリから外部メモリへ、の4通りの場合がある。

【００８４】また、本実施形態では、ＲＩＳＣ型のプロセッサを仮定しているので、データ転送を転送の要因別に分けるとすると、次のように場合分けされる。なお、転送の際に転送先が持つべき鍵ペア・タグの指定についても、示す。

【００８５】（１）プロセッサ・コア２０１の命令フェッチデコーダ２２０による命令フェッチの結果生じるデータ転送

このときの転送先の鍵ペア・タグは、現在実行中の実効鍵ペアのタグである。

【００８６】（２）プロセッサ・コア２０１のメモリ・アクセス部２６０でロード命令あるいはストア命令が実行された結果生じるデータ転送

このときの転送先の鍵ペア・タグは、現在実行中の実効鍵ペアのタグである。ただし、本発明では、MIPSのロード命令、ストア命令に加えて、新たに、転送先が持つべき鍵ペア・タグを命令のオペランドとして指定するタグ指定付きロード命令、ストア命令を追加する。これらの命令の命令フォーマットや、オペランドの指定方法については、任意である。

【００８７】（３）プロセッサ・コア２０１の算術論理演算部２５０で命令が実行された結果生じるデータ転送

RISC型のプロセッサにおいては、演算命令は、レジスタのみをデータ転送先とするので、この場合のデータ転送は、レジスタファイル２３０上のレジスタ間のデータ転送でしかあり得ない。転送先の鍵ペア・タグは、現在の実効鍵ペアのタグである。

【００８８】次に、本実施形態において、秘匿すべきデ

ータを保護するための機構について説明する。データの保護は、算術論理演算部250の演算オペランド・タグ判定部252、メモリ・アクセス部260のデータ転送タグ判定部262、および命令実行制御部223において実現される。いずれにおいても、共通したタグ判定規約に従って、データ転送の成立の可否が決定される。判定の基準に用いるものは、①データの転送元を特定する識別子、②転送するデータに付加された鍵ペア・タグ(「データタグ」と呼ぶ)、および③転送先の持つべき鍵ペア・タグ(「転送先タグ」と呼ぶ)である。

【0089】データ転送に伴うタグ判定の規約として最低限必要な規約を、以下にリストする。実施の形態によっては、さらに転送を否決する形で限定する規約を加えてもよい。また、データ転送に伴い、必要であれば暗号処理が施される。

【0090】(1)内部メモリ間のデータ転送であって、転送元が内部キャッシュであるときには、データタグと転送先のタグが一致しているときに限り、転送を許可する。データはそのまま転送され、転送先には、データタグが付加される。

【0091】(2)内部メモリ間のデータ転送であって、転送元が内部キャッシュではないときは、無条件にデータ転送を許可し、データタグを転送先のタグとする。

【0092】(3)内部メモリから外部メモリへのデータ転送にあたっては、無条件にデータ転送を許可する。このとき、データは、データタグが指定する鍵ペアの保持する暗号鍵(データ鍵)で暗号化される。すなわち、内部メモリから外部メモリへデータを転送する場合は、まず、データ暗号処理部601でデータタグが指定する鍵ペアの保持するデータ鍵を用いて暗号化される。命令フェッチは、読み出し専用であり、外部に書き出すという操作は存在しない。

【0093】(4)外部メモリから内部メモリへのデータ転送にあたっては、無条件にデータ転送を許可する。このとき、データは、転送先タグが指定する鍵ペアの保持する暗号鍵で復号化される。すなわち、外部メモリから内部メモリにデータを転送する場合は、それが命令フェッチに由来するデータ転送である時には、命令復号処理部501を経由して、転送先タグが指定する鍵ペアの保持するプログラム鍵を用いて復号化される。命令フェッチ以外では、データ暗号処理部601を経由して、転送先タグが指定する鍵ペアのデータ鍵を用いて復号化される。

【0094】(5)外部メモリ間のデータ転送には、本実施形態のプロセッサは関与しない。従って、外部メモリ間のデータ転送の挙動については、従来技術と同様である。

【0095】次に、このような保護機能を用いた処理の例として、オペレーティング・システム(OS)の基本的な処理であるコンテキスト・スイッチについて、図12～14を参照して説明する。前述の通り、例として用いるのは、MIPS Technologies社のR3000系プロセッサである。

【0096】前提条件として、ステップS1201に示すように、暗号化されたプログラムprogram-1がユーザ・モードで実行されている。そのプログラムを復号するためのプログラム鍵progkey-1と、この鍵の新規登録時に生成されたデータ鍵datakey-1とが、鍵ペアとして鍵ペア・テーブル710に格納されている。この鍵ペアを特定する鍵ペア・タグをtag-1とする。

【0097】ここで、外部からの割り込み等を理由として、プロセッサに例外が発生したとする。プロセッサは、ステップS1203に示すように、現在のプログラムカウンタの値を例外復帰用レジスタに退避させる。このとき、実効鍵ペア・タグも例外復帰用レジスタに退避される。ステップS1205に示すように、プロセッサの動作モードはユーザ・モードからカーネル・モードに遷移する。タグの値は実効鍵ペアのタグ値から、カーネル・モード用に予約されたタグ値tag-Kに切り替わる。動作モードとタグ値の切り替えによって、ステップS1207に示すように、OSの一部である例外処理ルーチンが起動される。例外処理ルーチンが終了したら、ステップS1209に示すように、再度ユーザ・モードに切り替わり、コンテキストが復帰する。

【0098】図13は、例外処理ルーチンS1207の詳細なステップを示す。まず、ステップS1301に示すように、例外発生時に実行していたプログラムのコンテキスト、すなわちレジスタファイル230の内容を、外部のメモリに格納するストア命令を実行する。さらに、ステップS1303で、退避した実効鍵ペアタグ(tag-1)が特定する鍵ペアを、鍵ペア管理部701からレジスタファイル230に読み出す(既存の鍵ペア読み出し操作)。なお、レジスタファイル230に読み出した鍵ペアデータは、OSのデータであるため、tag-Kが付加されている。ステップS1305で、鍵ペアデータをメモリに転送する。この一連の操作において、OSが不当にユーザモードのデータ(これにはtag-1が付加されている)に算術論理演算を行ったとすれば、それは前述した演算の制約条件から不許可となる。一方、ユーザモードでのデータを外部に転送する操作は、データ転送の制約条件で許可されるデータ転送である。この時点でユーザのコンテキストはレジスタファイル230の外に退避されたが、その退避先がデータキャッシュ上(内部メモリ)であるか、外部メモリにまで到達したかは、データキャッシュの状態に依存する。ただし、OSの動作という視点からは、レジスタファイル230からのコンテキストの退避は完了している。そして、ステップS1307で割り込み処理を実行する。

【0099】図14は、例外処理ルーチンが完了した後

の、ユーザのコンテキストの復帰処理(すなわちプログラムの再開) のフローを示す。まず、ステップS1401で、メモリに退避させておりた鍵ペアをOSのデータとしてメモリからロードする。ステップS1403で、この鍵ペアをtag-1に対応する鍵ペアとして鍵ペア・テーブル710に格納する (既存の鍵ペア書き込み操作)。ステップS1405で、メモリに退避してあったコンテキストをレジスタファイル230に復帰させる。このときに行うのは、転送先タグ付ロード命令である。転送先タグとしてはtag-1を指定する。さらに、ステップS1407で、例外復帰用レジスタに、復帰するプログラムカウンタの値と、実効鍵ペア・タグ (tag-1)とを格納する。最後に、ステップS1409で、例外復帰命令 (MIPS-IVではERET命令)を用いてユーザ・モードに遷移することで、コンテキスト復帰が完了する。この一連の操作でのデータ転送も、前述したデータ転送の規約条件で許可されるデータ転送である。

【0100】コンテキストの保存と復帰の際に、tag-1が付加されているデータが保護されていることは、以下のことからも確認される。まず、データ転送の規約条件により、tag-1以外の鍵ペア・タグを実効鍵ペア・タグとして実行しているときには、tag-1のデータは演算の対象となり得ない。一方、tag-1で特定されるプログラム鍵で復号されたプログラムの実行中は、実効鍵ペア・タグの値はtag-1であり、マイクロプロセッサ内部の処理について、OSは知ることができない。実行されていたデータが、割り込み等によって外部メモリへと転送されるときには、tag-1で特定される鍵ペアのデータ鍵で暗号化される。このデータ鍵は、プロセッサの秘密鍵を知ることなしに得ることは不可能である。また、コンテキスト復帰時には、タグ付きロード命令でデータを取り戻したが、このタグはOSの実効鍵ペア・タグとは異なるタグなので、OSからのアクセスは許可されない。逆に、OS自身の実効鍵を、復帰したユーザのコンテキストの鍵ペア・タグであるtag-1にすりかえたとしても、このときは、OSにとって知ることのできないプログラム鍵をもって、自らの命令列を復号することになり、予期できない命令を実行することになる。

【0101】このように、プログラム鍵とデータ鍵を鍵ペアとして不可分に扱うことにより、特権モードで実行するプログラムからも、秘密データを秘匿することが可能となる。

【0102】<第2実施形態>図15は、本発明の第2実施形態にかかるマイクロプロセッサで使用する鍵ペアテーブル810の構成図である。第1実施形態においては、一つのプログラム鍵と一つのデータ鍵とを1対1対応で関連付けて扱う方法について述べたが、第二の実施形態においては、一つのプログラム鍵に複数のデータ鍵を関連付けて扱う方法について述べる。

【0103】マルチタスク環境下では、同じひとつのプログラムについて、異なる種類のデータ処理がなされる場合もある。そのような場合は、プログラムの復号鍵 (プログラム鍵) はひとつであるが、実行処理されたデータを暗号化/復号化するためのデータ鍵は別々である。これら別々のデータ鍵のすべてについてプログラム鍵とのペアを作って格納しようとすると、膨大な記憶容量を要することになる。

【0104】そこで第2実施形態では、プログラム鍵のインデックスとデータ鍵のインデックスとをペアにして格納し、さらに、プログラム鍵およびデータ鍵をばらばらに格納しておく。

【0105】図15は、第2実施形態の鍵ペア・テーブル810の構成例を示す。鍵ペア・テーブル810は、鍵自体を保持する鍵テーブル820と、鍵を間接参照する形でインデックスで構成した鍵ペア参照テーブル830とで構成される。

【0106】鍵テーブル820は、プログラム鍵とデータ鍵とを個別にエントリした配列から構成される。一方、鍵ペア参照テーブル830は、鍵ペアを表わすものではあるが、鍵のペアを直接格納するのではなく、プログラム鍵のインデックス831-1と、データ鍵のインデックス831-2から構成される。これらのインデックスを用いて、ある特定のプログラムと、このプログラムによって実行される複数種類のデータとを特定することが可能になる。たとえば、図14の例では、インデックスを用いて、プログラム鍵#3で復号化された同じプログラムについて、データ鍵#4で暗号化/復号化されるデータと、データ鍵#5で暗号化/復号化されるデータとを、それぞれ組み合わせることができる。

【0107】鍵テーブル820の動作は、次の読み出し操作と、書き込み操作である。

【0108】(1) 鍵の読み出し
パラメータとして与えられるインデックスにより特定される鍵エントリの共通鍵が、鍵テーブル820から取り出される。

【0109】(2) 鍵の書き込み
まず、使用されていない鍵エントリを一つ割り当てる。その鍵エントリに、パラメータとして与えられる共通鍵 (プログラム鍵またはデータ鍵) を格納する。割り当てた鍵エントリを特定するインデックスを出力する。

【0110】鍵ペア・テーブル810の動作は、第1実施形態で述べた鍵テーブル710の動作と基本的に同じである。ただし、鍵ペア・テーブル810は、鍵テーブル820と鍵ペア参照テーブル830とに分割されていることから、鍵の読み出し、書き込みの詳細については、次のようになる。

【0111】(1) プログラム鍵の読み出し
命令復号処理部インタフェース703と接続されるポートを介する読み出し操作である。参照テーブル830のインデックスペアで特定される鍵のうち、プログラム鍵

のインデックスを用いて鍵テーブル８２０の鍵読み出し操作を行ない、その結果を出力する。

【０１１２】（２）データ鍵読み出し

データ暗号処理部インタフェース７０４と接続されるポートを介する読み出し操作である。参照テーブル８３０のインデックスペアで特定される鍵のうち、データ鍵のインデックを用いて鍵テーブル８２０の鍵読み出し操作を行ない、その結果を出力する。

【０１１３】（３）鍵ペアの読み出し

鍵ペア制御部７２０と接続されるポートを介する読み出し操作である。参照テーブル８３０のインデックスペアにより特定されるペアのプログラム鍵とデータ鍵の双方のインデックスを用いて、それぞれの鍵を鍵テーブル８２０から読み出し、得られた2つの鍵を出力する。

【０１１４】（４）鍵ペアの書き込み

鍵ペア制御部７２０と接続されるポートを介する書き込み操作である。パラメータとして与えられるプログラム鍵とデータ鍵とを、それぞれ鍵テーブル８２０の書き込み操作で鍵テーブル８２０に格納する。結果として得られる鍵エントリのインデックスを、参照テーブル８３０のプログラム鍵インデックスとデータ鍵インデックスに格納する。

【０１１５】鍵ペア・テーブル８１０は、鍵ペア制御部７２０から見て同じ動作を提供するインタフェースを備えている。したがって、鍵ペア制御部７２０の動作であるところの、新規鍵ペアの登録、既存の鍵ペアの読み出し、既存の鍵ペアの書き込み、の3つの操作は、第1実施形態で述べたものと同一である。

【０１１６】ただし、第2実施形態では、一つのプロセスが複数のデータ鍵を用いる手段を提供するために、鍵ペア・テーブル８１０と鍵ペア制御部７２０に、第1実施形態での操作の他に、次の操作を追加する。

【０１１７】まず、鍵ペア・テーブル８１０に追加する操作は、以下の通りである。

【０１１８】＊データ鍵の書き込み

鍵ペア制御部７２０と接続されるポートを介する書き込み操作であって、パラメータとして、プログラム鍵インデックスと、データ鍵と、鍵ペア・タグを受け取る。まず、データ鍵を、鍵テーブル８２０の書き込み操作で、鍵テーブル８２０に格納する。結果として得られる鍵エントリのインデックスと、パラメータとして受け取ったプログラム鍵インデックスとを、鍵ペア・タグがインデックスとして指定する鍵ペアのプログラム鍵インデックス８３１－1とデータ鍵インデックス８３１－2のそれぞれに格納する。

【０１１９】次に、鍵ペア制御部７２０に以下の操作を追加する。

【０１２０】＊新しいデータ鍵の登録

プロセッサ・コア２０１からは、データ鍵を所有するプロセスを特定する第1の鍵ペア・タグと、新たに登録するデータ鍵を特定するための第2の鍵ペア・タグと、登録するデータ鍵を第1の鍵ペア・タグのプログラム鍵で暗号化した鍵データとが提示される。鍵データは、鍵ペアデータではないが、暗号化鍵ペア・レジスタ７２６に格納される。

【０１２１】まず、第1の鍵ペア・タグをインデックスとして、鍵テーブル８２０の鍵ペアの読み出し操作を行い、鍵ペアを鍵ペア・レジスタ７２４に読み出す。

【０１２２】鍵ペア暗号処理部７２５は、暗号化鍵ペア・レジスタ上のデータを、暗号化された鍵とみなして、鍵ペア・レジスタ７２４のプログラム鍵７２４－1を用いて復号化し、その結果を鍵ペア・レジスタ７２４のデータ鍵７２４－2に格納する。

【０１２３】鍵ペア制御部7２０は、第1の鍵ペア・タグを利用して読み出したプログラム鍵のインデックスと、鍵ペア・レジスタ７２４のデータ鍵７２４－2とを、鍵ペアテーブル８１０のデータ鍵書き込み操作を用いて、鍵ペアとして登録する。

【０１２４】この操作により、登録したデータ鍵のインデックスと、そのデータ鍵とともに用いられるプログラム鍵のインデックスとを一組にした新しい鍵インデックスのペアが作成される。

【０１２５】言い換えれば、複数の鍵ペアを、プログラム鍵が共通であることによりグループ化する操作である。しかも、グループに追加する操作においては、プログラム鍵による暗号処理を必要とするため、プログラム鍵を知るプロセスにのみ可能なものである。

【０１２６】上述のデータ鍵登録操作は、プログラム鍵で暗号化したデータとして与えるものであるが、プログラム鍵ではない別の鍵、たとえばプロセッサの公開鍵あるいは別のデータ鍵で暗号化したデータとして与えるという変形も可能である。このときは、プログラム鍵に基づいたグループ化ではなく、別の鍵をもとにしたグループ化となり、プログラム間で鍵を共有する状況に適用できる。

【０１２７】第1実施形態において、データ転送および演算操作に対して、鍵ペア・タグの値が一致することを、その操作が成功するための条件する制約機構について述べた。第2実施形態においては、異なる鍵ペア・タグであっても、同一のプロセスが生成した鍵ペアであれば、データ転送や演算操作を許可するものとする。すなわち、データ転送や演算操作の成立条件判定に、第1実施形態での鍵ペア・タグの値そのものに加え、その鍵ペア・タグが指定する鍵ペアの、プログラム鍵インデックスとデータ鍵インデックスを条件判定に用いる。

【０１２８】第1実施形態で例示した、オペレーティング・システムによるコンテキスト・スイッチの実施例は、第2実施形態においても同様に動作する。

【０１２９】第1および第2実施形態を、ともに機能を実現する要素をブロック図を用いて説明したが、これら

23

は機能の論理的な分割方法を示したものであり、機能ブロックのプロセッサ上での物理的な配置を示すものではない。例えば、実施形態の説明で鍵ペアは一つのテーブルに一組にして保持するものとしたが、プロセッサ上の物理的な配置としては、プログラム鍵を集めたテーブルは命令復号処理部の近くに、データ鍵を集めたテーブルはデータ暗号処理部の近くに配置する、といった物理的な構成方法を除外するものではない。

【０１３０】

【発明の効果】以上説明したように、本発明によれば、プログラムを復号化するためのプログラム鍵と、データを暗号化するためのデータ鍵とが、暗号学的に不可分のものとしてプロセッサ内部で扱われるため、プログラムを実行する主体であるプロセスを、オペレーティング・システムを介することなく、プロセッサが保護することが可能となる。したがって、他のユーザ・プログラムのみならず、オペレーティング・システムからも、プログラムの持つ秘密情報を守ることができる。

【０１３１】また、本発明によれば、プロセッサによる保護の対象であるプロセスを識別するためのタグをプロセッサ内部のデータに付加することにより、保護の対象となるデータを、復号化した状態で内部メモリに保持したまま、プロセスの切替えを行なうことができる。

【図面の簡単な説明】

【図1】本発明の第1実施形態に係るマイクロプロセッサの全体構成例を示す図である。

【図2】本発明の基本的な処理フローのうち、データを暗号化して退避させる流れを示す図である。

【図3】本発明の基本的な処理フローのうち、退避させたデータを復帰させる流れを示す図である。

【図4】図1に示すプロセッサ・コアの構成図である。

【図5】図4に示す算術論理演算部の詳細図である。

【図6】図4に示すシステムレジスタの詳細図である。

【図7】図1に示す命令キャッシュの図である。

【図8】図1に示すデータキャッシュの図である。

24

【図9】図1に示す命令暗号処理部の図である。

【図10】図1に示すデータ暗号／復号処理部の図である。

【図11】図1に示す鍵ペア管理部の図である。

【図12】本発明のマイクロプロセッサにおける割り込み発生時の処理フローを示す図である。

【図13】図12の例外処理ルーチンの詳細を示す図である。

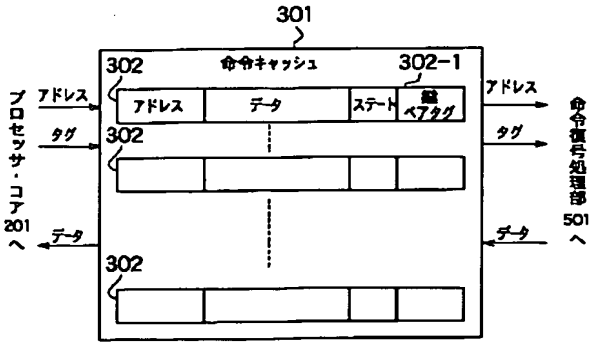
【図14】図12のコンテキスト復帰ステップの詳細を示す図である。

【図15】本発明の第2実施形態にかかる鍵ペア・テーブルの図である。

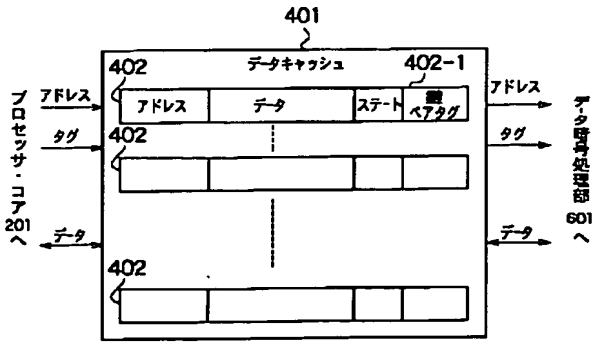
【符号の説明】

１０１　マイクロプロセッサ
１０２　プロセッサ・バス
１０３　外部バス・インタフェース
２０１　プロセッサ・コア
２１０　システム・レジスタ群（第3の記憶領域）
２３０　レジスタ・ファイル（第2の記憶領域）
２５０　算術論理演算部
２５２　演算オペランド・タグ判定部
２６０　メモリ・アクセス部
２６２　データ転送タグ判定部
２７０　鍵ペア管理部インタフェース
３０１　命令キャッシュ
４０１　データ・キャッシュ
５０１　命令復号処理部
６０１　データ暗号／復号処理部
７０１　鍵ペア管理部
７１０、８１０　鍵ペア・テーブル（第1の記憶領域）
７２０　鍵ペア制御部
７２５　鍵ペア暗号化部
７２６　暗号化鍵ペア・レジスタ
８２０　鍵テーブル
８３０　鍵ペア参照テーブル

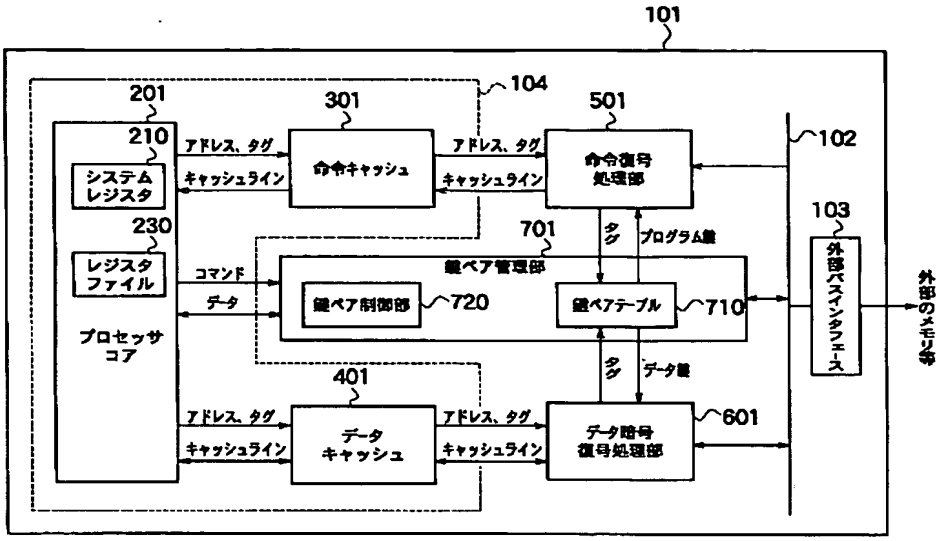
【図7】



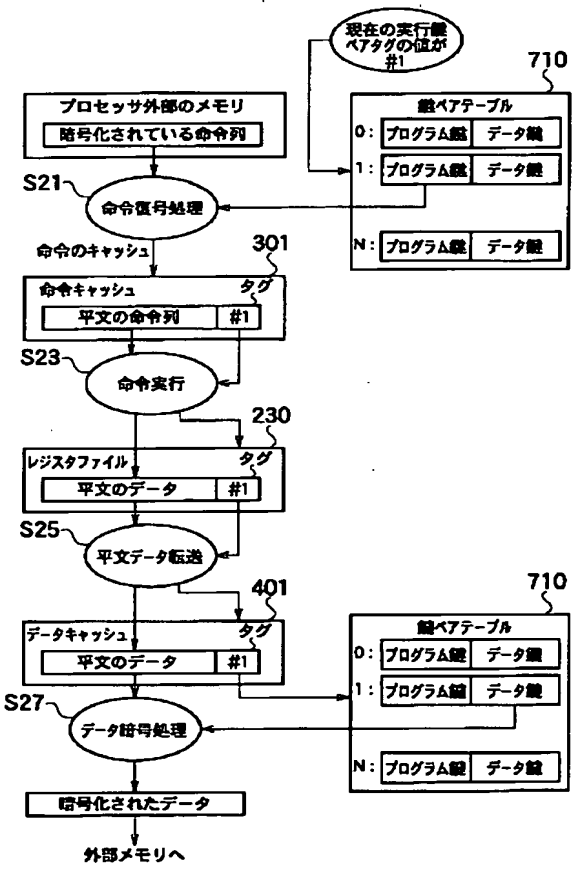
【図8】



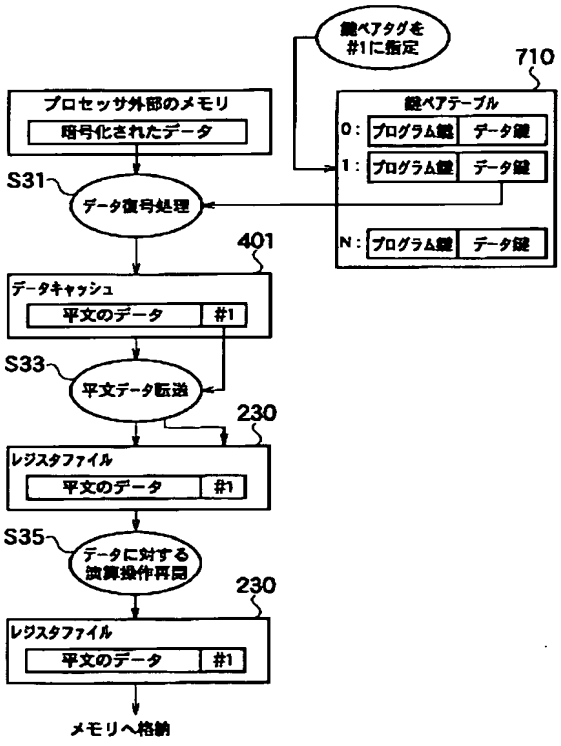
【図1】


101
104
201
210
システムレジスタ
230
レジスタファイル
プロセッサコア
301
命令キャッシュ
501
命令復号処理部
102
103
外部バスインタフェース
外部のメモリ等
アドレス、タグ
キャッシュライン
コマンド
データ
701
鍵ペア管理部
鍵ペア制御部
720
鍵ペアテーブル
710
タグ
プログラム鍵
データ鍵
401
データキャッシュ
601
データ暗号復号処理部

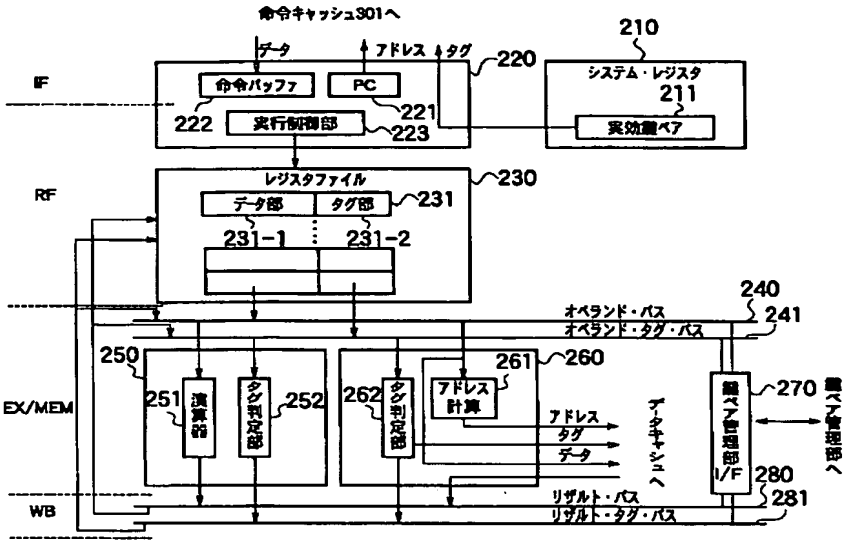
【図2】


現在の実行鍵ペアタグの値が#1
710
プロセッサ外部のメモリ
暗号化されている命令列
鍵ペアテーブル
0：プログラム鍵 データ鍵
1：プログラム鍵 データ鍵
N：プログラム鍵 データ鍵
S21
命令復号処理
命令のキャッシュ
301
命令キャッシュ
タグ
平文の命令列
#1
S23
命令実行
230
レジスタファイル
平文のデータ
S25
平文データ転送
401
データキャッシュ
S27
データ暗号処理
暗号化されたデータ
外部メモリへ

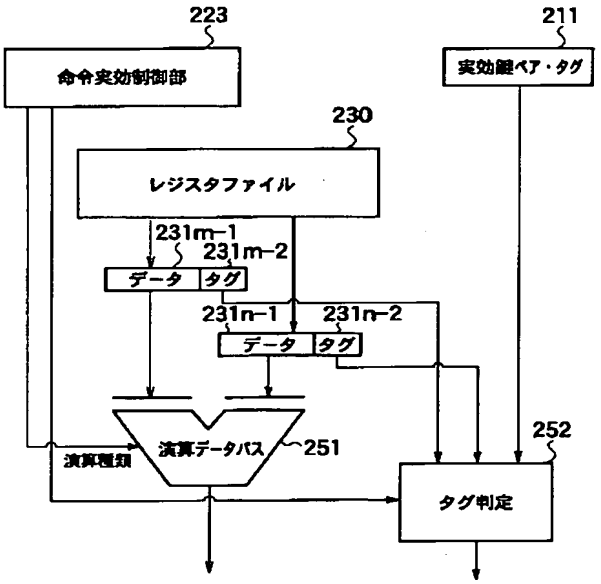
【図3】


鍵ペアタグを#1に指定
710
プロセッサ外部のメモリ
暗号化されたデータ
鍵ペアテーブル
0：プログラム鍵 データ鍵
1：プログラム鍵 データ鍵
N：プログラム鍵 データ鍵
S31
データ復号処理
401
データキャッシュ
平文のデータ
#1
S33
平文データ転送
230
レジスタファイル
S35
データに対する演算操作再開
メモリへ格納

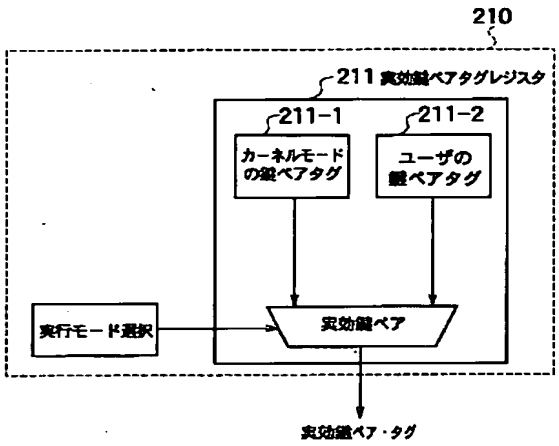
【図4】


命令キャッシュ301へ
データ
アドレス
タグ
220
210
システム・レジスタ
211
実効鍵ペア
IF
命令バッファ
PC
222
実行制御部
221
223
RF
レジスタファイル
230
データ部
タグ部
231
231-1
231-2
240
オペランド・バス
241
オペランド・タグ・バス
250
251
演算部
タグ判定部
252
262
タグ判定部
アドレス計算
261
260
EX/MEM
アドレス
タグ
データ
データキャッシュへ
鍵ペア管理部I/F
270
鍵ペア管理部へ
280
281
WB
リザルト・バス
リザルト・タグ・バス

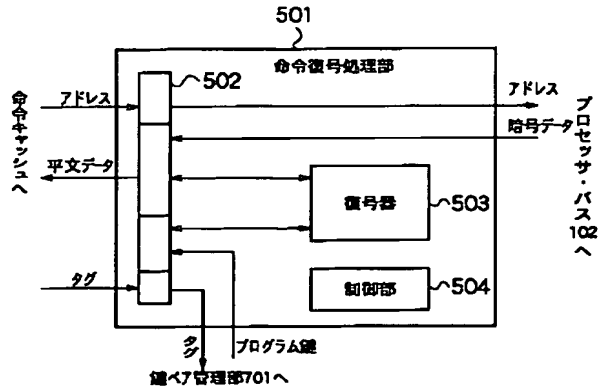
【図5】


223
命令実効制御部
211
実効鍵ペア・タグ
230
レジスタファイル
231m-1
231m-2
データ
タグ
231n-1
231n-2
データ
タグ
演算種類
演算データパス
251
252
タグ判定

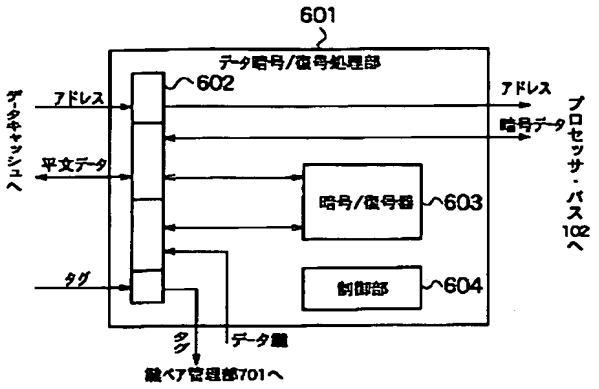
【図6】


210
211 実効鍵ペアタグレジスタ
211-1
211-2
カーネルモードの鍵ペアタグ
ユーザの鍵ペアタグ
実行モード選択
実効鍵ペア
実効鍵ペア・タグ

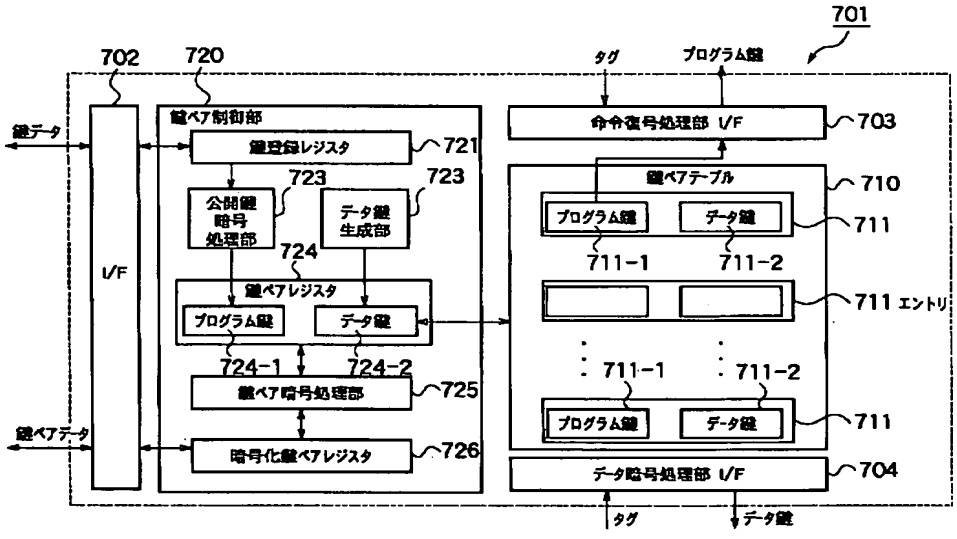
【図9】


501
命令復号処理部
502
アドレス
命令キャッシュへ
平文データ
タグ
アドレス
暗号データ
プロセッサ・バス102へ
復号器
503
制御部
504
タグ
プログラム鍵
鍵ペア管理部701へ

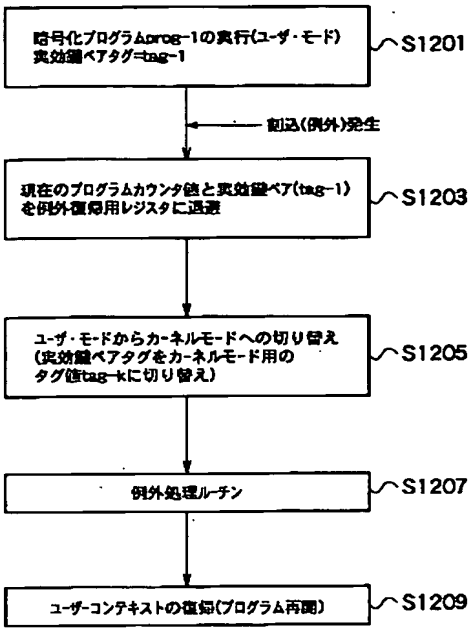
【図１０】


601
データ暗号/復号処理部
602
アドレス
平文データ
タグ
暗号/復号器
603
制御部
604
アドレス
暗号データ
プロセッサ・バス102へ
データキャッシュへ
タグ
データ鍵
鍵ペア管理部701へ

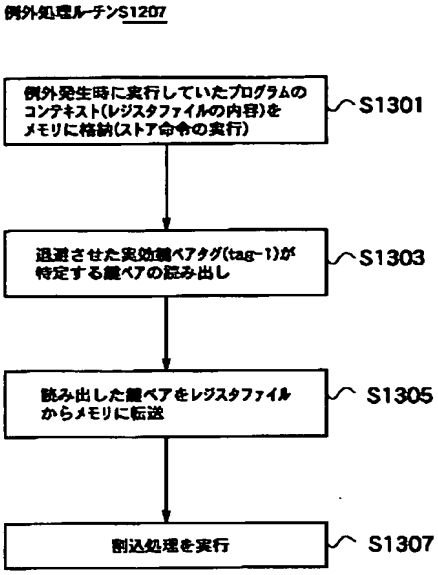
【図１１】


701
702
720
鍵データ
I/F
鍵ペア制御部
鍵登録レジスタ
721
723
公開鍵暗号処理部
723
データ鍵生成部
724
鍵ペアレジスタ
プログラム鍵
データ鍵
724-1
724-2
鍵ペア暗号処理部
725
鍵ペアデータ
暗号化鍵ペアレジスタ
726
タグ
プログラム鍵
命令復号処理部 I/F
703
鍵ペアテーブル
710
プログラム鍵
データ鍵
711
711-1
711-2
711 エントリ
711-1
711-2
プログラム鍵
データ鍵
711
データ暗号処理部 I/F
704
タグ
データ鍵

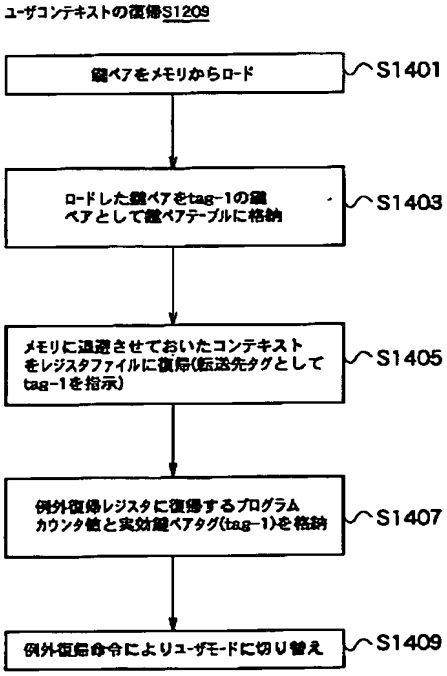
【図１２】


暗号化プログラムprog-1の実行(ユーザ・モード)
実効鍵ペアタグ=tag-1
S1201
割込(例外)発生
現在のプログラムカウンタ値と実効鍵ペア(tag-1)を例外復帰用レジスタに退避
S1203
ユーザ・モードからカーネルモードへの切り替え(実効鍵ペアタグをカーネルモード用のタグ値tag-kに切り替え)
S1205
例外処理ルーチン
S1207
ユーザーコンテキストの復帰(プログラム再開)
S1209

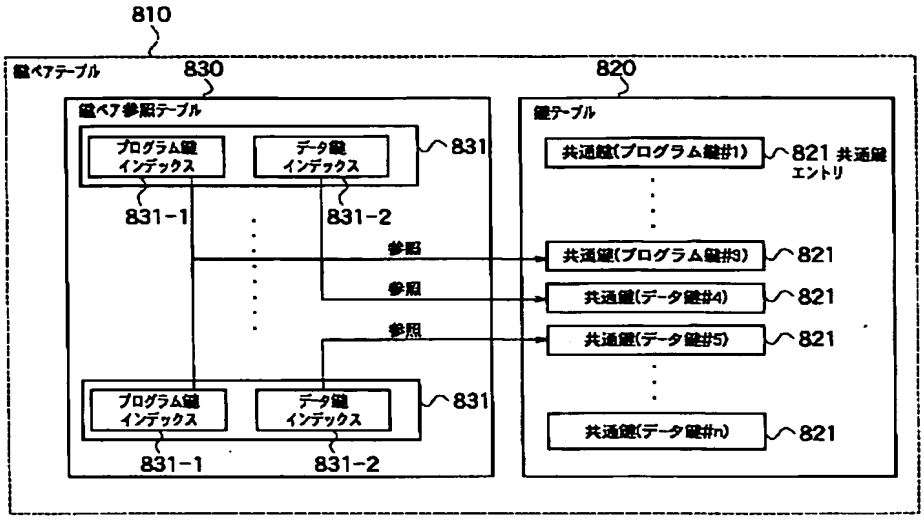
【図１３】


例外処理ルーチンS1207
例外発生時に実行していたプログラムのコンテキスト(レジスタファイルの内容)をメモリに格納(ストア命令の実行)
S1301
退避させた実効鍵ペアタグ(tag-1)が特定する鍵ペアの読み出し
S1303
読み出した鍵ペアをレジスタファイルからメモリに転送
S1305
割込処理を実行
S1307

【図１４】


ユーザコンテキストの復帰S1209
鍵ペアをメモリからロード
S1401
ロードした鍵ペアをtag-1の鍵ペアとして鍵ペアテーブルに格納
S1403
メモリに退避させておいたコンテキストをレジスタファイルに復帰(転送先タグとしてtag-1を指示)
S1405
例外復帰レジスタに復帰するプログラムカウンタ値と実効鍵ペアタグ(tag-1)を格納
S1407
例外復帰命令によりユーザモードに切り替え
S1409

【図１５】



フロントページの続き

(72)発明者　寺本　圭一
神奈川県川崎市幸区小向東芝町1 番地　株式会社東芝研究開発センター内
(72)発明者　尾崎　哲
神奈川県川崎市幸区小向東芝町1 番地　株式会社東芝研究開発センター内
(72)発明者　藤本　謙作
神奈川県川崎市幸区小向東芝町1　株式会社東芝研究開発センター内

Ｆターム(参考)　5B017 AA03 BA07 CA15 CA16
5B076 FA13 FC08
5B098 DD01
5J104 AA01 AA16 EA02 EA04 EA25
JA21 NA02